UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/038,338	10/23/2001	Arun P. Gupta	SUNMP024	7269

25920 7590 07/12/2006

MARTINE PENILLA & GENCARELLA, LLP
710 LAKEWAY DRIVE
SUITE 200
SUNNYVALE, CA 94085

EXAMINER
TRAN, QUOC A

ART UNIT	PAPER NUMBER
2176	

DATE MAILED: 07/12/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/038,338	**Applicant(s)** GUPTA, ARUN P.	
	Examiner Quoc A. Tran	**Art Unit** 2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 April 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12,14-17 and 19-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12, 14-17 and 19-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to Amendment filed on 04/14/2006, to the original application filed 10/23/2001.

2. Claims 1-12, 14-17 and 19-21 are pending. Applicants amended claims 1, 6, 11, 16 and 21 cancelled claims 13 and 18. Claims 1, 11,16 and 21 are independent claims.

3. The Examiner withdrawn 35 U.S.C. 101 rejection of claim 21.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> ***(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.***

5. **Claims 1-12, 14-17 and 19-21** are rejected under 35 U.S.C. 103(a) as being unpatentable by Mason US006826716B2 – filed 09/26/2001 (hereinafter Mason), in view of view of Davis US 20050198042A1-Division of No. 09/573,413 filed 05/18/2000 (hereinafter Davis), further in view of Washburn et al. US005157779A – issued 10/20/1992 (hereinafter).

In regard to independent claim 1, executing a computer software test application on a platform (as taught by Mason at col. 15, lines 25-35, a further object of the present invention is to provide a test program for testing J2EE application on a flat form, such as test generator in the J2EE (Java 2 Enterprise Edition), the terms J2EE application/web application is used here in the broadest sense to encompass the executing a computer software test application);

generating test result in results in an Extensible Markup Language (XML) enabled format (as taught by Mason at col. 15, lines 25-35, a further object of the present invention is to provide a test program for testing J2EE application on a flat form, such as test generator in the J2EE (Java 2 Enterprise Edition), wherein the web application via the XML deployment descriptors, and a test generator could generate tests to explicitly related QOS (Quality-Of – Service) of an Enterprise application. The test generator output Java code, wherein the Java xml parser was employed for breaking down element to usable parts (as taught by Mason at col. 2, lines 10-20). The above schema was used in the broadest sense to encompass the claim limitation, such as test resulting in xml enable format).

Mason does not explicitly teach, **and using the XML enabled test results to create a test summary report**, however (Davis at the Abstract, discloses methods and systems provide a "chart view" for a markup language referred to as Reusable Data Markup Language ("RDML"). Generally, a chart view comprises the components necessary for automatically manipulating and displaying a graphical display of numerical data contained in RDML markup documents. RDML is a markup language, such as the Hypertext Markup Language ("HTML") or the Extensible Markup Language ("XML"), also Davis at page 9, paragraph [0115], discloses RDSL style sheets 106 act as templates for output reports. The RDML data object (discussed below) in the RDML data viewer 100 can be placed into a report using one or more different style sheets. RDSL, a fully compliant implementation of XSL, allows a data publisher to provide multiple report formats for its data),

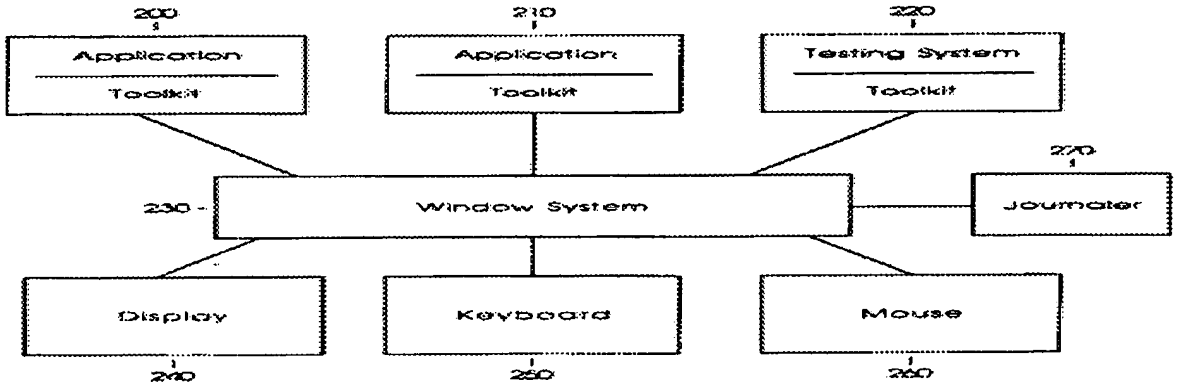
wherein the XML enabled test results is capable of being rearranged, however (Davis at page 5 paragraphs [0067] –[0073], provide a markup language, referred to as Reusable Data Markup Language ("RDML"), and a data viewer referred to as the RDML data viewer that is used to retrieve, manipulate and view documents in the RDML format *(rearranged)*. Generally, RDML permits the browsing and manipulation of numbers, and allows the viewer to act as a combination Web browser and spreadsheet/analytic application that may automatically read numbers from multiple online sources, understand their meaning, and manipulate them without human intervention. Similarly, Reusable Data Style Language ("RDSL") style sheets 106, another optional input to the data viewer 100, can be applied to data documents to create specially-formatted output reports. A RDSL is a fully compliant implementation of Extensible Style Language ("XSL") which is described in detail in "XML Bible," Elliotte Rusty Harold, IDG Books Worldwide, 1999),

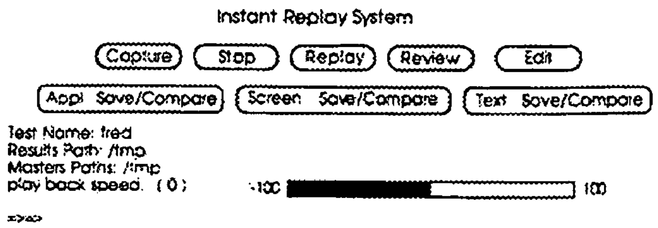
the rearranged XML enabled test results..., independent of an XML format capability of the platform, however (Davis at page 5 paragraphs [0067] –[0073], provide a markup language, referred to as Reusable Data Markup Language ("RDML"), and a data viewer referred to as the RDML data viewer that is used to retrieve, manipulate and view documents in the RDML format. Generally, RDML permits the browsing and manipulation of numbers, and allows the viewer to act as a combination Web browser and spreadsheet/analytic application that may automatically read numbers from multiple online sources, understand their meaning, and manipulate them without human intervention. Similarly, Reusable Data Style Language ("RDSL") style sheets 106, another optional input to the data viewer 100, can be applied to data documents to create specially-formatted output reports. A RDSL is a fully compliant implementation of Extensible Style Language ("XSL") which is described in detail in "XML Bible," Elliotte Rusty Harold, IDG Books Worldwide, 1999, further Davis at page 26 paragraph [0303] through page 27 paragraph [0321], includes a means of utilizing the tagging wizard for creating a tagged text document in RDML, and also Davis at page 9, paragraph [0115], discloses RDSL style sheets 106 act as templates for output reports. The RDML data object (discussed below) in the RDML data viewer 100 can be placed into a report using one or more different style sheets. RDSL, *a fully compliant implementation of XSL, allows a data publisher to provide multiple report formats for its data*),

It is noted that a "chart view" for a markup language referred to as Reusable Data Markup Language ("RDML"). Generally, a chart view comprises the components necessary for automatically manipulating and displaying a graphical display of numerical data contained in RDML markup documents. RDML is a markup language, such as the Hypertext Markup Language ("HTML") or the Extensible Markup Language ("XML" - **XML format capability)** (see Davis at the Abstracts).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified teaching of Mason, wherein the test program was perform on the web application producing the out put in Java code enabling the java xml for parsing the out put result into usable parts, to include a means for report generator interfaces in a markup language format such as XML format capability (it is noted that XML format capability is tags language (markup language) that provide the capability of tagging the test result), of Davis's teaching. One of the ordinary skills in the art would have been motivated to perform such a modification to reduces the expense, time, and complexity of data manipulation by addressing the aforementioned problems of documentation of data, non-standardization of analytic routines, and low conceptual-level calculations of data and encapsulating data and its documentation together in machine-readable form that can be used interactively (as taught by Davis at page 5 paragraph [0074]).

Mason and Davis do not explicitly teach, **the computer software test application having one or more test suites, each test suit including one or more test cases, each test case configured to test an aspect of the platform, each test suite tag encapsulating the test results corresponding to each test suite of the computer software test application, including test suite tags.** However, Washburn teaches an automated testing system that capable of testing a wide variety of software programs across different type of hardware architectures (test suit and platform), and provides flexible testing parameters and resulting the journal of status of a test (i.e. PAST and/or FAIL) to a particular computer software application (test case) (see Washburn at col. 1, line 50 though col. 2, line 15). also (supports in Fig. 2 items 200, 210, 220, 230 and 270 and Fig. 11b),

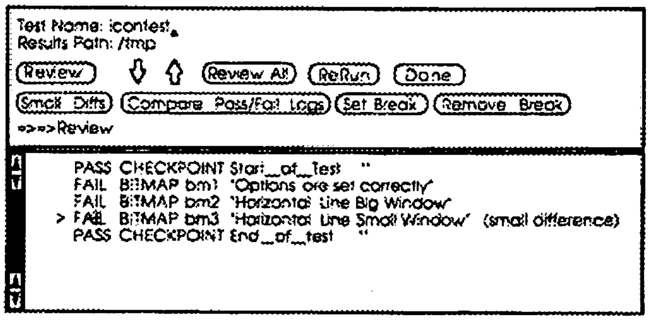




Test Name: fred
Results Path: /tmp
Masters Paths: /tmp
play back speed. (0)

=>=>

Figure 9a



Test Name: icontest
Results Path: /tmp

=>=>Review

```
   PASS  CHECKPOINT Start__of__Test   ''
   FAIL  BITMAP bm1  'Options are set correctly'
   FAIL  BITMAP bm2  'Horizontal Line Big Window'
 > FAIL  BITMAP bm3  'Horizontal Line Small Window'  (small difference)
   PASS  CHECKPOINT End__of__test   ''
```

Figure 9b

It is noted that in Fig. 2 and 9a-b, wherein a plurality of applications 200, 210 operate in a window system environment through the window system 230 (test suit and platform), (see Washburn at col. 5, lines 5-10). Further Washburn teaches the window system 230 includes the journaling mechanism item 270 (test summary report) (see (see Washburn at col. 5, lines 15-20). Further Washburn teaches the window system 230 includes the journaling mechanism item 270 (test summary report) (see (see Washburn at col. 5, lines 15-20). the review module is selected through a menu item which is part of the main menu (FIG. 9a) to bring up the review module menu (FIG. 9b) of the test result that represent differences in the actual data Pass or Fail (test case) (see Washburn at col. 12, lines 20-35).

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Washburn's automated testing system that capable of testing a wide variety of software programs across different type of hardware architectures (test suit and platform), and provides flexible testing parameters and resulting the journal of status of a test (i.e. PAST and/or FAIL) to a particular computer software application (test case) to Mason and Davis's testing program and system, providing an automated testing system that capable of testing a wide variety of software programs across different type of hardware architectures (test suit and platform), and provides flexible testing parameters and resulting the journal of status of a test (i.e. PAST and/or FAIL) to a particular computer software application (test case) and to reduces the expense, time, and complexity of data manipulation by addressing the aforementioned problems of documentation of data, non-standardization of analytic routines, and low conceptual-level calculations of data and encapsulating data and its documentation together in machine-readable form that can be used interactively (as taught by Davis at page 5 paragraph [0074]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply a "chart view" for a markup language referred to as Reusable Data Markup Language ("RDML"). Generally, a chart view comprises the components necessary for automatically manipulating and displaying a graphical display of numerical data contained in RDML markup documents. RDML is a markup language, such as the Hypertext Markup Language ("HTML") or the Extensible Markup Language ("XML" - **XML format capability)** (see Davis at the Abstracts) to Washburn's automated testing system that capable of testing a wide variety of software programs across different type of hardware architectures (test suit and platform), and provides flexible testing parameters and resulting the journal of status of a test (i.e. PAST and/or FAIL) to a particular computer software application (test case) providing an automated testing system that capable of testing a wide variety of software programs across different type of hardware architectures (test suit and platform), and provides flexible testing parameters and resulting the journal of status of a test (i.e. PAST and/or FAIL) to a particular computer software application (test case) and to reduces the expense, time, and complexity of data manipulation by addressing the aforementioned problems of documentation of data, non-standardization of analytic routines, and low conceptual-level calculations of data and encapsulating data and its documentation together in machine-readable form that can be used interactively (as taught by Davis at page 5 paragraph [0074]).

In regard to independent claim 11, incorporate substantially similar subject matter as cited in claim 1 above, and in further view of the following, and is similarly rejected along the same rationale,

a parser that processes a test execution log file a logical parser that processes the well-formed XML test reports file to produce a logically arranged XML test reports file (as taught by Mason at col. 1, line 65 through col. 2, line 5, provide a mechanism for selecting application behaviors at assembly or deployment time to generate a well-formed XML test reports file),

Mason does not explicitly teach, **a logical parser operable to use the well-form XML test report file,** however (Davis at page 8 paragraphs [0101] through page 9 paragraph [0115], also see Fig. 2-7discloses an automated test, performing data analytics while the data is in a An

RDML document server 218 functions when RDML documents 102 are being created dynamically. The server 218 queries the existing database 230 for the desired line items, queries the image database 226 for documentation items and instructions for constructing the RDML document 102, and finally creates a valid, well-formed RDML document, wherein RDSL style sheets 106 act as templates for output reports. The RDML data object (discussed below) in the RDML data viewer 100 can be placed into a report using one or more different style sheets. RDSL, a fully compliant implementation of XSL, allows a data publisher to provide multiple report formats for its data) Examiner read the above in the broadest reasonable interpretation to the claim limitation, wherein log file would have been an obvious variant of report file to a person of ordinary skill in the art at the time the invention was made,

converting the logically arranged XML test reports file into an HTML test summary report, however (Davis at page 6 paragraphs [0080], discloses RDML utilizes and manipulates data, that automatically normalizes, collates, transforms, and formats the data (XML into HTML).

wherein the test execution log file includes XML statements generated independent of an XML format capability. However, It is noted that a "chart view" for a markup language referred to as Reusable Data Markup Language ("RDML"). Generally, a chart view comprises the components necessary for automatically manipulating and displaying a graphical display of numerical data contained in RDML markup documents. RDML is a markup language, such as the Hypertext Markup Language ("HTML") or the Extensible Markup Language ("XML" - **log file includes XML statements - XML format capability)** (see Davis at the Abstracts). Further more Davis teaches RDML document editor 220 permits users to edit the actual elements and attributes of an RDML, RMML, or RDSL document (102, 104 and 106). The documents may be presented in a tree view for selection and direct editing. A text-based window may display the contents of the selected document for editing. Besides basic editing, it performs a number of utility functions: (1) search and replace, (2) validation, (3) *well-formedness testing*, (4) hyperlink validation, (5) cut-and-paste of elements, and (6) replacement of elements with defaults (**the test execution)** (see Davis at page 11 paragraph [0127]).

It is noted that in Fig. 2 and 9a-b, wherein a plurality of applications 200, 210 operate in a window system environment through the window system 230 (test suit and platform), (see Washburn at col. 5, lines 5-10). Further Washburn teaches the window system 230 includes the journaling mechanism item 270 (test summary report) (see (see Washburn at col. 5, lines 15-20). Further Washburn teaches the window system 230 includes the journaling mechanism item 270 (test summary report) (see (see Washburn at col. 5, lines 15-20). the review module is selected through a menu item which is part of the main menu (FIG. 9a) to bring up the review module menu (FIG. 9b) of the test result that represent differences in the actual data Pass or Fail (test case) (see Washburn at col. 12, lines 20-35).

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Washburn's automated testing system that capable of testing a wide variety of software programs across different type of hardware architectures (test suit and platform), and provides flexible testing parameters and resulting the journal of status of a test (i.e. PAST and/or FAIL) to a particular computer software application (test case) to Mason and Davis's testing program and system, providing an automated testing system that capable of testing a wide variety of software programs across different type of hardware architectures (test suit and platform), and provides flexible testing parameters and resulting the journal of status of a test (i.e. PAST and/or FAIL) to a particular computer software application (test case) and to reduces the expense, time, and complexity of data manipulation by addressing the aforementioned problems of documentation of data, non-standardization of analytic routines, and low conceptual-level calculations of data and encapsulating data and its documentation together in machine-readable form that can be used interactively (as taught by Davis at page 5 paragraph [0074]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply a "chart view" for a markup language referred to as Reusable Data Markup Language ("RDML"). Generally, a chart view comprises the components necessary for automatically manipulating and displaying a graphical display of numerical data contained in RDML markup documents. RDML is a markup language, such as the Hypertext Markup Language ("HTML") or the Extensible Markup Language ("XML" - **XML format capability)** (see Davis at the Abstracts) to Washburn's automated testing system that capable of testing a wide variety of software programs across different type of hardware architectures (test suit and platform), and provides flexible testing parameters and resulting the journal of status of a test (i.e. PAST and/or FAIL) to a particular computer software application (test case) providing an automated testing system that capable of testing a wide variety of software programs across different type of hardware architectures (test suit and platform), and provides flexible testing parameters and resulting the journal of status of a test (i.e. PAST and/or FAIL) to a particular computer software application (test case) and to reduces the expense, time, and complexity of data manipulation by addressing the aforementioned problems of documentation of data, non-standardization of analytic routines, and low conceptual-level calculations of data and encapsulating data and its documentation together in machine-readable form that can be used interactively (as taught by Davis at page 5 paragraph [0074]).

In regard to independent claim 16, incorporate substantially similar subject matter as cited in claims 1 and 11 above, and is similarly rejected along the same rationale.

In regard to independent claim 21, incorporate substantially similar subject matter as cited in claims 1 and 11 above, and in further view of the following, and is similarly rejected along the same rationale,

Examiner read the above in the broadest reasonable interpretation to the claim limitation, wherein log file, a test ID and data identifying a test suite to which a test belongs would have been an obvious variant of report file and a mapping dictionary based on text, class sets, and user input, to a person of ordinary skill in the art at the time the invention was made.

In regard to claims 2-4, 6-9, 14 and 19 incorporate substantially similar subject matter as cited in claims 11 and 16 above, and are similarly rejected along the same rationale.

In regard to claim 5, incorporate substantially similar subject matter as cited in claims 16 above, and further in view of the following, and are similarly rejected along the same rationale,

document type definition (DTD), however (as taught by Davis at page 4, paragraph [0034]),

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified teaching of Mason, wherein the test program was perform on the web application producing the out put in Java code enabling the java xml for parsing the out put result into usable parts, to include a means for report generator interfaces in a markup language format such as XML and DTD, of Davis's teaching. One of the ordinary skills in the art would have been motivated to perform such a modification to reduces the expense, time, and complexity of data manipulation by addressing the aforementioned problems of documentation of data, non-standardization of analytic routines, and low conceptual-level calculations of data and encapsulating data and its documentation together in machine-readable form that can be used interactively (as taught by Davis at page 5 paragraph [0074]).

In regard to claim 10, incorporate substantially similar subject matter as cited in claims 16 above, and further in view of the following, and are similarly rejected along the same rationale,

includes links to failure description pages, wherein the failure description pages provide a detailed description of a particular test failure (as taught by Mason at col. 7, lines 30-45, since XML is self describing formatted data that required a family of technologies is defined in relation to extend functionality in relation to XML, such as "Xlink," was intended to describe a standard way to add hyperlinks to XML files in collaborating with the rejection set forth above in claim 16. Examiner reads the above schema in the broadest sense to encompass the xml linking to the failure description pages).

In regard to claim 15, incorporate substantially similar subject matter as cited in claims 10 above, and are similarly rejected along the same rationale.

In regard to claims 12, 17 and 20 incorporate substantially similar subject matter as cited in claims 16 and 5 above, and are similarly rejected along the same rationale.

Response to Arguments

6. Applicant's arguments filed 04/14/2006, with respect to the rejection(s) of claim(s) 1-12, 14-17 and 19-21, have been considered but are moot in view of the new ground(s) of rejection. To address the amended portions the Examiner introduces the Washburn reference as for the un-amended portions have been considered, but they are not persuasive. The reason is set forth in the current Office Action cited above and further view of the following:

Brief description of cited prior arts:

Mason discloses a test program for testing (Java 2 Enterprise Edition) J2EE application on a flat form, such as test generator in the J2EE (Java 2 Enterprise Edition) comprise modules, the testing including identifying (204), from an application deployment descriptor, modules comprised within the J2EE application, wherein the web application via the XML deployment descriptors, and a test generator could generate tests to explicitly related QOS (Quality-Of – Service) of an Enterprise application. The test generator output Java code, wherein the Java xml parser was employed for breaking down element to usable parts (as taught by Mason at the Abstract and at col. 2, lines 10-20).

Davis discloses an automated testing tool for programming language dependencies and data type inconsistencies from multiple sources (Davis at page 7 paragraph [0085] through page 8 paragraph [0092]).

Washburn discloses an automated testing system that capable of testing a wide variety of software programs across different type of hardware architectures, and provides flexible testing parameters and resulting the journal of status of the test (i.e. PAST and/or FAIL) to a particular computer software application (see Washburn at col. 1, line 50 though col. 2, line 15).

Response to Arguments:

Beginning on page 7 of the Remarks/Arguments (hereinafter the remarks), Applicant argues the following issues, which are accordingly addressed below.

Applicant's arguments, on page 7 of the remarks that Mason in view of Davis fails to teach the amended portions,

The examiners have been considered but are moot in view of the new ground(s) of rejection (see Washburn reference as for the un-amended portions in the rejection above for detail).

Applicant's arguments, on pages 7-9 of the remarks that Mason in combination with Davis fail to teach fail to teach software test application, the computer software test application having one or more test suites, and generating test result in results in an Extensible Markup Language (XML) enabled format;

The examiner respectfully disagrees. The examiner respectfully notes that Mason discloses a test program for testing (Java 2 Enterprise Edition) J2EE application on a flat form, such as test generator in the J2EE (Java 2 Enterprise Edition) comprise modules, the testing including identifying (204), from an application deployment descriptor, modules comprised within the J2EE application, wherein the web application via the XML deployment descriptors, and a test generator could generate tests to explicitly related QOS (Quality-Of –Service) of an Enterprise application. The test generator output Java code, wherein the Java xml parser was employed for breaking down element to usable parts (as taught by Mason at the Abstract and at col. 2, lines 10-20).

(as taught by Mason at col. 15, lines 25-35, a further object of the present invention is to provide a test program for testing J2EE application on a flat form, such as test generator in the J2EE (Java 2 Enterprise Edition), wherein *the web application via the XML deployment descriptors* (via XML in an information retrieval system- as taught by Mason at col. 2, lines 10-20).

further more in the current office action states above, the Examiner respectfully introduces Washburn reference to address those amend portions and further supporting and curing the deficiency of Mason and Davis. Washburn teaches an automated testing system that capable of testing a wide variety of software programs across different type of hardware architectures, and provides flexible testing parameters and resulting the journal of status of the test (i.e. PAST and/or FAIL) to a particular computer software application (see Washburn at col. 1, line 50 though col. 2, line 15).

Therefor the Examiner the rejection of independent claims 1, 11, 16 and 215 and theirs dependencies are appropriated for at least the reason cited above at this time.

Conclusion

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is (571) 272-4103. The examiner can normally be reached on Monday through Friday from 9 AM to 5 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Herndon R. Heather can be reached on (571) -272-4136. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Quoc A, Tran
Patent Examiner
Technology Center 2176
June 30, 2006

William L. Bashore
WILLIAM BASHORE
PRIMARY EXAMINER